

# Sabancı University

## Faculty of Engineering and Natural Sciences

CS406/CS531 Parallel Computing / Parallel Processing and Algorithms

Spring 2019-2020

Homework #1

Due: 01/03/2020 - 23:55

### PLEASE NOTE:

Your program should be a robust one such that you have to consider all relevant programmer mistakes and extreme cases; you are expected to take actions accordingly!

You **HAVE TO** write down the code on your own.  
You **CANNOT HELP** any friend while coding.  
Plagiarism will not be tolerated!!

## 1 Introduction

In this assignment, you will implement a sequential Futoshiki Solver. The definition of the Futoshiki Puzzle and techniques for solving could be found at [futoshiki.org](http://futoshiki.org). The purpose of the homework is implementing an algorithm for Futoshiki Puzzle **as efficient as possible**. Note that the grading will be done based on the end-to-end execution times of implementations.

## 2 Program Flow

You will be reading puzzles from provided files with provided reader and calculate the solution of the given puzzle. The filename of the matrix must be passed to your program by command line (use argc/argv). The format of the input files will be as follows:

- Each input will begin with an integer  $n$  that indicates the size of the matrix.
- Each input will contain a square matrix which cells with values presented by positive numbers and empty cells presented with  $-1$ 's. The values are separated with whitespaces.
- Each input will have constraints for the puzzle after the matrix. One line for one constraint. Constraints are given in a format that; the value of cell resides in coordinate of first two numbers is bigger than the cell resides in latter two. For example, constraint 1 2 1 3 means that cell at (1,2) will have bigger value than cell at (1,3). Therefore an input `input1.txt` will be in the format:

4

2 -1 -1 -1

-1 -1 -1 -1

-1 -1 -1 -1

-1 -1 -1 1

1 2 1 3

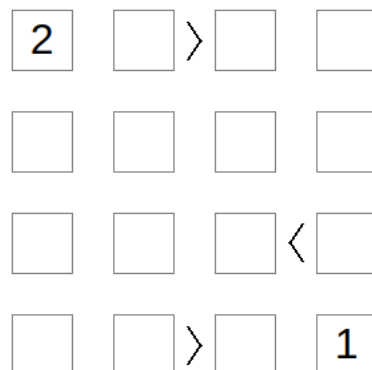
3 4 3 3

4 2 4 3

will represent the matrix given right.

(Notice 1-based index.)

The input files will be given in correct format  
and you do not need to implement any input checks.



- The result will be printed out to standard output as a matrix which columns separated with empty space and rows separated with `\n`.
- For example, solution to the given puzzle should be printed out to matrix as:

```

2 4 1 3
3 1 4 2
1 2 3 4
4 3 2 1

```

### 3 Solution Calculation

In order to calculate the solution of the given puzzle, you will need to develop an efficient algorithm and correctly implement it using `C++`. In this homework, no algorithm or hint will be provided. But you can check the correctness of your algorithm via provided solutions for each input file. Moreover, you can benchmark your implementation's efficiency by comparing it's run-time to our example implementation's run-time on nebula computer:

File	Size	Time (second)
input1.txt	4x4	0.0002
input1_2.txt	4x4	0.0001
input2.txt	5x5	0.0013
input2_2.txt	5x5	0.0188
input_3.txt	6x6	0.0098
input3_2.txt	6x6	0.1126
input_4.txt	7x7	1.29
input_4_2.txt	7x7	4.04
input_5.txt	8x8	82.06
input_5_2.txt	8x8	154.19

Your report will have equal importance as the implementation. A report of at least two pages should be submitted along with the code, including:

- A general explanation of the implementation
- Execution times
- Tricks done for efficiency (providing spatial/temporal locality, preprocessing etc.)
- How you compile and run your program.

### 4 Some Remarks

In the grading process, three things will be checked:

- Correctness of your implementation
- Efficiency of your implementation
- How well the report is written

Your code will be compiled with the `-O3` optimization flag, so you are expected to include the relevant results in your project. You can use the given example matrices and its solution to check the correctness of your code. The solutions of the matrices given for homework will also be provided.

## 5 What and Where to Submit (PLEASE READ, IMPORTANT)

**Please don't forget to submit your code and the report together.** Your **REPORT** must be a **pdf** file (preferable prepared by L<sup>A</sup>T<sub>E</sub>X but MS Word converted pdf's are also OK). It must contain the description of the optimizations you implemented, i.e., it must explain how you improved the performance, what was the timings before and after. You must do this for both of the -O0 and -O3 optimization options. Please see above what else do you need to include in the report.

The grading process is not automatic. However, the students are expected to strictly follow the guidelines in order to have a smooth grading process. If you do not follow these guidelines, depending on the severity of the problem created during the grading process, 5 or more penalty points are to be deducted from the grade. The name of your source code file that contains your program must be **futoshiki\_hw1.cpp**. Similarly the report must be named as **report\_hw1.pdf**.

Put both of these files into a folder named

***SUCourseUserName\_YourLastname\_YourName\_HWnumber***

Your SUCourse user name is actually your SUNet username that is used for checking sabanciuniv e- mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SUCourse user name is cago, name is Çağlayan, and last name is Özbugsizkodyazaroglu, then the folder name must be:

***cago\_Caglayan\_Ozbugsizkodyazaroglu\_hw1***

Do not add any other character or phrase to the folder name. Make sure that it contains the last version of the source code and the report. Compress this folder using a zip program. Please use "zip" compression. **"rar" or another compression mechanism is NOT allowed. Please make sure that you include both of the files in the compressed folder.**

You will receive no credits if your compressed folder does not expand or it does not contain the correct files. The name of the zip file should be as follows:

***SUCourseUserName\_YourLastname\_YourName\_HWnumber.zip***

For example zubzipler.Zipleroglu.Zubeyir\_hw1.zip is a valid name, but

***hw1\_hoz\_HasanOz.zip, HasanOzHoz.zip***

are **NOT** valid names. **Submit via SUCourse ONLY!** You will receive no credits if you submit by other means (e-mail, paper, etc.).

Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.

Good Luck!

CS406-531 Team (Fatih Taşyaran, Kamer Kaya)